Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2024 SCPE. Volume 25, Issues 2, pp. 1018–1028, DOI 10.12694/scpe.v25i2.2502

A CLASS SPECIFIC FEATURE SELECTION METHOD FOR IMPROVING THE PERFORMANCE OF TEXT CLASSIFICATION

VENKATESH V[*], SHARAN S B[†], MAHALAXMY S[‡], S MONISHA[§], ASHICK SANJEY D S[¶], AND ASHOKKUMAR P[‖]

Abstract. Recently, a significant amount of research work has been carried out in the field of feature selection. Although these methods help to increase the accuracy of the machine learning classification, the selected subset of features considers all the classes and may not select recommendable features for a particular class. The main goal of our paper is to propose a new class-specific feature selection algorithm that is capable of selecting an appropriate subset of features for each class. In this regard, we first perform class binarization and then select the best features for each class. During the feature selection process, we deal with class imbalance problems and redundancy elimination. The Weighted Average Voting Ensemble method is used for the final classification. Finally, we carry out experiments to compare our proposed feature selection approach with the existing popular feature selection methods. The results prove that our feature selection method outperforms the existing methods with an accuracy of more than 37%.

Key words: Feature selection, machine learning, class-specific feature selection.

1. Introduction. The textual data volume is increasing day by day. This causes the need for a text classification that categorizes the text into two or more classes. Increasing text classification performance involves many stages such as feature selection, feature weighting, instance selection, and instance weighting [1]. Feature selection is the process of picking a subset of features that maximizes the performance of a machine-learning model. Feature weighting is a standard to assign a score to each feature based on its importance. Instance selection involves identifying proper instances that help the machine learning model to learn the relationship between the features and class much more easily. Similar to feature weighting, instance weighting assigns an importance score to each instance. This paper focuses on feature selection.

Feature selection is one of the important methods which is applied to the text classification problem. In feature selection, a subset of features is selected which can be used in the classification models which outputs an increased accuracy [2]. Much research shows that only a handful of features are enough to build a good classification model rather than the entire set of features. There are lots of advantages when the feature selection is applied which include dimension reduction, increased classification performance, removal of redundancy, and so on [3].

Feature selection aims to remove the useless features and retain only the precise features that help the classifier to understand the relationship between the features and classes [4]. The feature selection methods can be categorized into filter types, wrapper types, and embedded types. The filter-based feature selection method replies to statistical information and selects the best subset of features. Whereas the wrapper method leverages the performance of feature classification by selecting the optimal subset of features. Unlike filter methods, the wrapper methods are classifier-dependent, which means if the classification model changes, then the subset

[*]Department of Cybersecurity and Internet of Things, Sri Ramachandra Faculty of Engineering and Technology, Sri Ramachandra Institute of Higher Education and Research

[†]Department of Cybersecurity and Internet of Things, Sri Ramachandra Faculty of Engineering and Technology, Sri Ramachandra Institute of Higher Education and Research

[‡]Department of Artificial Intelligence and Data Analytics, Sri Ramachandra Faculty of Engineering and Technology, Sri Ramachandra Institute of Higher Education and Research

[§]Department of Cybersecurity and Internet of Things, Sri Ramachandra Faculty of Engineering and Technology, Sri Ramachandra Institute of Higher Education and Research

[¶]Department of Cybersecurity and Internet of Things, Sri Ramachandra Faculty of Engineering and Technology, Sri Ramachandra Institute of Higher Education and Research

[‖]Department of Artificial Intelligence and Machine Learning, Sri Ramachandra Faculty of Engineering and Technology, Sri Ramachandra Institute of Higher Education and Research

of selected features is also changed. Embedded methods are the mix of filter and wrapper methods [5]. The feature selection method used in this manuscript is based on filter-based feature selection.

Filter methods are proven to be must faster and provide more advantages over the rest of the two types [6]. As in text classification, the number of features is very high thus creating a high dimensional space with many zeros in the feature matrix. The filter-based methods help to reduce the dimensions and make the classification model more efficient. Moreover, the filter-based methods are powerful in removing inconsistencies such as noisy data, and redundancies between features [7]. In this paper, we make use of filter-based feature selection to extract a promising feature subset.

Feature selection can be general or class-specific. In the first case, the feature selection algorithm picks a set of features that represents the entire class as a whole, whereas the second type of feature selection helps to pick features that are specific to each class [8]. The selected features may not be the same for all the classes, hence the advantage of class-specific feature selection is, that the model can learn the unique aspects of a class.

Based on the literature, the following limitations are found in the text classification domain.

- The existing feature selection methods do not capture unique features to represent a class uniquely
- The imbalance dataset classification remains a challenge in the field of text classification
- The problem of merging multiple binary classifications into a single multi-class classification needs to be addressed.

We propose a novel class-specific feature selection method with multiple levels of classification to improve the text classification performance.

In this manuscript, we develop a unique class-specific feature selection method that helps the machine learning model to understand the relationship between the input and output variables much deeper. The use of class-specific feature selection comes with its challenges such as imbalanced instances during the training phase, difficulty in ensemble the result of classification, and so on [10] [9]. We focus on optimizing the class-specific feature selection methods in text classification by the use of sampling and ensemble voting. The main contributions of the paper are as follows.

1. To propose a text classification model that has strong knowledge about all the classes.
2. To eliminate the class imbalance problem by using sampling and similarity-based elimination.
3. To increase the performance of the machine learning classification by introducing ensembling methods.

The remainder of this paper is organized in the following manner. Some recent related works regarding the class-specific feature selection are reviewed in section 2. The section 3 explains the problem statement of this manuscript. Section 4 elaborates the proposed classification model on class-specific features. Section 5 compares the proposed approach with the popular machine learning models by using a standard benchmark dataset. Section 6 ends this paper by providing the conclusion and scheduling a few future works.

2. Related Works. A four-stage framework which includes binarization, balancing, feature extraction, and finally, classification was used by [11]. The authors have used four machine learning models namely kNN (k Nearest Neighbour), SVM (Support Vector Machine), NB (Naive Bayes), and XGBoost for classification. Furthermore, eight feature selection methods have been used in their experiment. The dataset is split into training and testing by k fold method where k varies between 1 and 10. The results show that when k=10, all the model produces good performance with the maximum accuracy of 98.84 %.

A deep learning framework was proposed by [12]. In their work, the medical image classification is done using class-specific feature extraction methods. The final image classification is done by combining multiple convolution encoder-decoder frameworks. All the mini models learn about a discriminative, non-redundant training set. An 86.73 % accuracy was obtained by their proposed work.

A feature weighting method introduced by [13] uses the NB model for the classification as it is easy to use in text domains because the features are independent of each other. The authors suggested considering four conditions to measure the degree of relevance. The first one is the frequency of the term in the training dataset. The second one is the frequency of the term in the specific class. The third one represents the distribution of the class which contains the term and finally, the number of classes that contain the term. An accuracy of 86.05 % was obtained using their proposed algorithm.

A work by [14] uses class-specific feature extraction and principal component analysis for classification. They extend the traditional dictionary learning for all classes to specific classes. Each class has its internal dictionary to express more about itself. Eight datasets are used to validate their proposed work.

In some cases, the features of a dataset cannot be used to classify the instances into one or more classes, in that case, the dataset is termed an uncertain dataset. To classify an uncertain dataset, rough-based features are used. The authors in [15] use the kNN model to extract class-specific features from an uncertain dataset using rough set theory.

Building a classification model that works with high-dimensional data is very challenging. A work by [16] uses the J48 classifier to reduce the dimensions of the data and then extract the class-specific features for each class using the term weighting approach. The authors show that the method runs much faster than the traditional methods.

A work proposed by [17] extracts class-specific features that automatically determine the number, length, and start values. This work reduces the number of calculations required by the traditional models and thus increases the speed of calculations. Since the features are extracted for each class separately, all these processes can be done parallelly to reduce the time.

The authors in [18] present a class-specific feature extraction model which samples a small number of training patterns from the original data, then for each class in the dataset, the algorithm extracts the patterns. The authors introduce a new classification algorithm that uses all the extracted class features to determine the classes for the instances.

Another interesting work by [19] uses three metrics to extract class-specific feature subsets. The first metric is the correlation between the features and the target class. The next metric is the relevance of the features in the selected subset and the last metric is the redundancy of the features in the selected subset.

3. Problem Statement. Let the text corpus D represented by N number of documents as shown in Eq 3.1. The goal is to classify each of the N documents to one of the classes c_i where $c_i \in C$

$$D = \{d_1, d_2, ..., d_N\} \tag{3.1}$$

Each unique document in D is represented as a tuple of features and classes as presented in Eq 3.2.

$$D = (x_i, c)^M \tag{3.2}$$

The goal is to perform class binarization which creates unique sub-sets of features for each class which can be expressed as Eq 3.3.

$$FS_i = \begin{cases} (x,1)\ \forall x \in X & ,if\ y = i \\ (x,0)\ \forall x \in X & ,if\ y \neq i \end{cases} \tag{3.3}$$

$$F_c = argmax_{f_i} I(f,c) \tag{3.4}$$

After the feature sets are extracted, the subset is used to train the machine learning models and predict the outcome as shown in Figure in 3.1. The candidate features are identified for each class, these features represent the class's unique features which can be the best for binary tests as denoted by Eq 3.4. To increase the efficiency, three machine learning model is used in each binary classification, and the weighted ensemble method is used to merge the result. The final classification is done by performing a mode ensemble approach.

4. The Class Specific Feature Extraction Approach. Feature extraction is a proven method to optimize the classification accuracy of a machine learning model, however, when the feature extraction is done on the entire dataset, the extracted subset of features may not represent the uniqueness of any one or more classes. This problem is overcome by a class-specific feature selection algorithm, where the features are extracted for each class and the subsets of two different classes may not be the same.

The first step is class binarization where we convert an N-class classification problem into an N two-class classification model (that is, we have only two classes- the positive and the negative). We start this process by splitting all the instances into two categories for each class, in one category, all the instances of the respective class are placed. In the other category, the rest of the entire instances are placed as shown in Figure 4.1. The main problem that is faced in this method is that, in each classification category, a huge imbalanced dataset is created. To eliminate the imbalance problem, we provide the following two solutions.

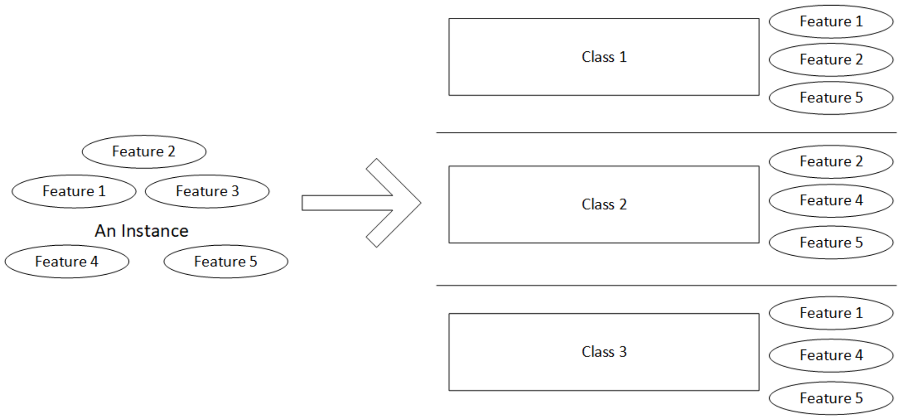

Fig. 3.1: The class-specific feature extraction process

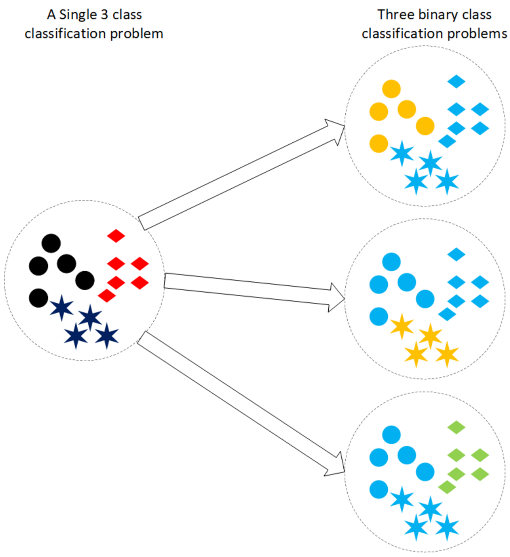

Fig. 4.1: The process of class binarization

1. Delete the similar instances in the majority class (the negative class)
2. Create instances using SMOTE (Synthetic Minority Oversampling Technique)

4.1. Feature Selection. Feature Selection plays an important role in data classification as it can shrink the dimension in the feature space and remove the redundancies. In this paper, we make use of three feature selection methods namely Information Gain (Eq 4.1), Gini Index (Eq 4.2), and Chi-Square (Eq 4.3). Information Gain is a popular feature selection method that measures how much information is provided by a feature to determine the target class. The higher the information, the more powerful the feature is. Once the information gained for all the features is found, then the top k features are selected and used to train the machine learning model.

$$H(x) = -\sum_i P_X(X_i) log(P_X(X_i)) \tag{4.1}$$

$$Gini = \sum_{i=1}^{m} P(t \mid c_i)^2 P(c_i \mid t)^2 \tag{4.2}$$

$$Chi = \frac{N * (tp * tn - fp * fn)}{(tp + fp) + (tp + fn) + (tn + fp) + (tn + fn)} \tag{4.3}$$

4.2. Deletion of Majority Instances. As the N number of classes is mapped into two classes of size 1 (positive) and N-1 classes (negative), the negative class size becomes very large. This creates a huge imbalanced dataset, thus making the classification model learn more about one class and less about another class. To create a balanced dataset, one approach is to delete the instance from the majority instance (the negative class). The deletion of instances should be done with proper care so that the important instance should not be deleted and also the exact relationship should be learned from the dataset. To implement the deletion, first, the most similar instances have to be identified in the feature space. Then just by keeping one of the similar instances, all other instances can be deleted. The main advantage of the above-mentioned method is that random deletion is prevented.

Similar instances can be grouped by using any distance matching method that satisfies the following properties(assume Θ is the distance finding algorithm).

1. $\Theta(X, Y)$ is always equal to $\Theta(Y, X)$
2. $\Theta(X, X)$ is always 0.
3. $\Theta(X, Y)$ is always > 0.

The similarity finding equation is presented at Eq 4.4. The function returns all the instances whose distance is less than a small threshold value Δ.

$$Similiar(X) = \{\forall x, x \in U, \Theta(X, x) <= \Delta\} \tag{4.4}$$

Once similar instances for each instance are found, we have many sets of instances. In each set, the instances inside them reflect the same spatial properties in the feature space, so it is safe to delete the entire members of the set by keeping only one. This ensures that the model can learn all the relationships between the input features and the target classes in the dataset.

Algorithm 1 Local Classification

Input: Instances with all features
Output: The class associated with it
$F_1 \leftarrow Features\ selected\ by\ eq$ 4.1
$F_2 \leftarrow Features\ selected\ by\ eq$ 4.2
$F_3 \leftarrow Features\ selected\ by\ eq$ 4.3
Let M_1 , M_2 *and* M_3 be three classification models build on F_1, F_2, *and* F_3 respectively
$y = W_{F_1} * M_{F_1} + W_{F_2} * M_{F_2} + W_{F_3} * M_{F_3}$
return y

4.3. Instance Generation using SMOTE. In this manuscript, we use two methods to bring an imbalanced dataset into a balanced dataset, the first one is to delete similar instances (of the majority class) in the feature space. The other method is to generate synthetic instances. The most popular method used in generating synthetic instances is SMOTE (Synthetic Minority Oversampling) [20]. Many problems like overfitting, and loss of important information are addressed in SMOTE. In recent years, many optimal versions of SMOTE are available [21].

One of the main advantages of the SMOTE is, that it does not generate random instances which creates an information loss in the training phase of the classification model. The SMOTE picks a random sample in the minority class and finds the k closest neighbors using the Euclidean distance finding formula. Then a new location is identified within the selected instances from the Euclidean formula and a new synthetic instance is created.

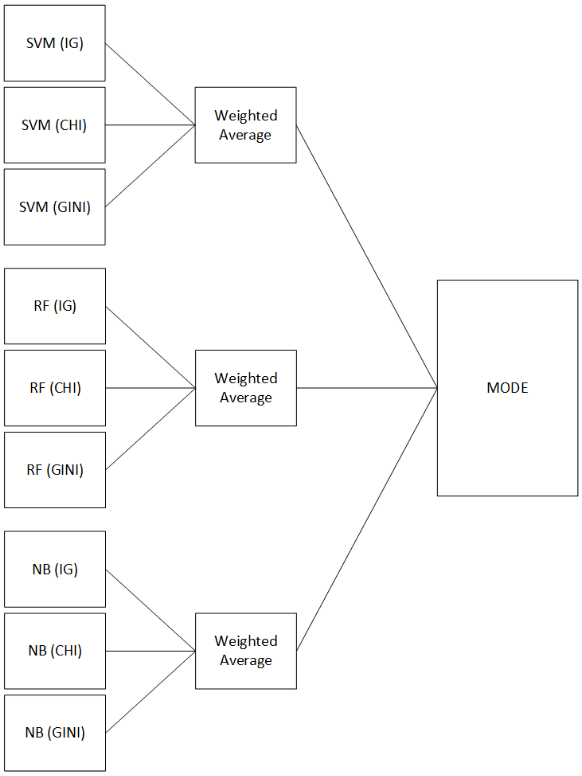

Fig. 4.2: The working of the classification model

4.4. Machine Learning Based Classification. We have used three machine learning models namely SVM, RF, and NB to classify the text documents and assign a class to them. We have adapted ensemble methods at multiple levels to increase the performance of the classification. We have selected these three machine learning models because of some reasons which we have explained in the following subsections.

4.4.1. Support Vector Machines. The SVM classifier can classify both linear and non-linear data. The SVM creates a non-linear hyperplane that can separate the instances. The main challenge an SVM model faces is to place the hyperplane in a perfect position that acts as a decision boundary that partitions the data instances. In text classification, the SVM works better as the feature space has many dimensions [22].

4.4.2. Naive Bayes. Naive Bayes is one of the statistical-based machine learning models that operate on the Bayes theorem. The semantics and underlying relationship between the words can be easily captured by using Naive Bayes. Since the classifier depends on conditional probabilities, it can be suited for a large dataset and gives optimal results.

4.4.3. Random Forest. Random Forest is one of the ensemble classifiers which combines multiple decision trees. Since multiple decision trees are used, each tree can be used to work on a feature. This ensures that the irrelevant features are not passed to other trees. The power of RF is the ability to generalize multiple trees.

4.4.4. Multi Level Ensemble Methods. For each class, we have each of these three machine learning models deployed three times one for each feature selection method. The classification results are finalized for each machine learning model based on the weighted average method as described in Eq 4.5. The full working of the ensemble model is described in algorithm 1. In the algorithm, the unique features for all the classes are extracted using three feature selection methods, and then the three classifiers such as SVM, NB, and RF are used to calculate the target class of the instance. The weighted voting ensemble method is used to find the

Table 5.1: Dataset Description

Sno	Dataset Name	# of classes	# of documents
1	Reuters-21578	10	9980
2	IMDB	2	50K
3	20newsgroup	20	18K

Table 5.2: Hyper parameters of ML model

Machine Learning Model	Hyperparameters
SVM	C=100, kernel = rbf
RF	estimators = 50

final class.

$$y = W_{IG} * M_{IG} + W_{CHI} * M_{CHI} + W_{GINI} * M_{GINI} \tag{4.5}$$

After each of the weighting average results are obtained the results are ensembled using majority ensembling as shown in Eq 4.6. y_1, y_2, and y_3 represent the outputs of the three weighted average methods. The abovementioned process is repeated parallelly for all the classes and the final class is determined by one hot ensembling. The working of the proposed approach is displayed in Figure 4.2.

$$y = mode(y_1, y_2, y_3) \tag{4.6}$$

5. Results and Discussion. We have experimented to test our proposed approach with three datasets. The details of the dataset are exhibited in Table 5.1.

In this experiment, we have used 10-fold validation. In this method, the dataset is randomly divided into 10 subsets. The proposed classifier is trained and tested 10 times where in each iteration, one subset is used as testing, and the remaining 9 subsets are used as training. The experiments are carried out in an i9 CPU with 16GB RAM. The details of hyper-parameters used in the experiment are shown in Table 5.2.

All the documents in the three datasets are pre-processed in the following order. First, the stop words are removed from the documents. Second, each word is stemmed and finally, all the features are represented in the bag of words model.

To evaluate the proposed approach, we have used four metrics known as accuracy, precision, and, recall. The metrics are calculated by using the formulas mentioned in Eq 5.1, 5.2 and, 5.3 respectively.

$$ACC = (TP + TN)/(TP + TN + FP + FN) \tag{5.1}$$

$$P = (TP)/(TP + FP) \tag{5.2}$$

$$R = (TP)/(TP + FN) \tag{5.3}$$

Table 5.3 provides the performance evaluation of all the machine learning models along with our proposed approach. It can be noted that the proposed approach outperforms all the existing classification models in the comparison. In existing models, NB produces the highest accuracy and recall whereas the SVM produces the highest precision.

From Table 5.4, it can be found that among all the classifiers in the comparison, the proposed approach exhibits the highest performance and has an advantage over the other machine learning models in the IMDB dataset. In the existing models, NB yields the highest performance in terms of accuracy, precision, and recall.

Table 5.3: Performance Evaluation of 20newspaper dataset

Model	Acc	Prec	Rec
kNN	0.75	0.8	0.75
RF	0.76	0.74	0.75
SVM	0.76	0.81	0.75
NB	0.77	0.80	0.77
LR	0.54	0.55	0.54
Proposed	**0.83**	**0.86**	**0.77**

Table 5.4: Performance Evaluation of IMDB

Model	Acc	Prec	Rec
kNN	0.62	0.75	0.67
RF	0.7	0.71	0.71
SVM	0.75	0.8	0.79
NB	0.76	0.81	0.75
LR	0.61	0.65	0.64
Proposed	**0.85**	**0.79**	**0.77**

Table 5.5: Performance Evaluation of Reuters Dataset

Model	Acc	Prec	Rec
kNN	0.81	0.81	0.8
RF	0.81	0.81	0.81
SVM	0.84	0.85	0.84
NB	0.85	0.85	0.85
LR	0.79	0.78	0.8
Proposed	**0.9**	**0.9**	**0.89**

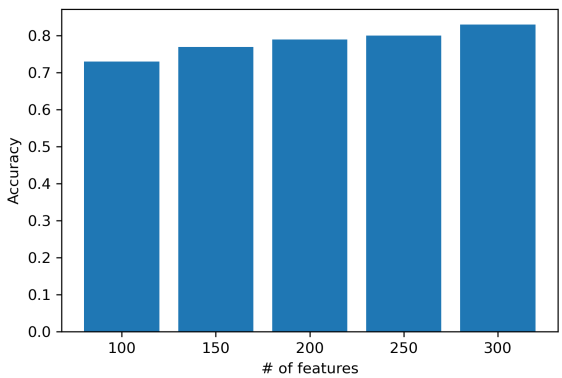

Fig. 5.1: The accuracy comparison of our proposed approach with respective to various number of features - 20newsgroup

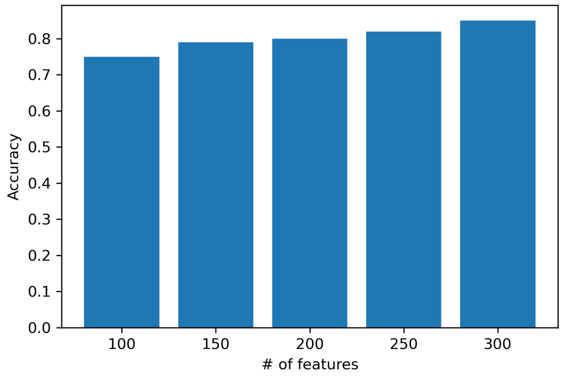

Fig. 5.2: The accuracy comparison of our proposed approach with respective to various number of features - IMDB

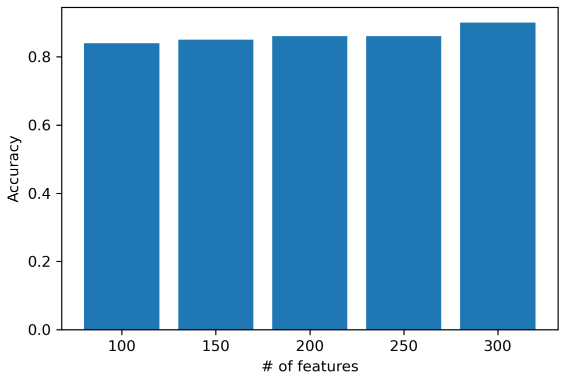

Fig. 5.3: The accuracy comparison of our proposed approach with respective to various number of features - Reuters-21578

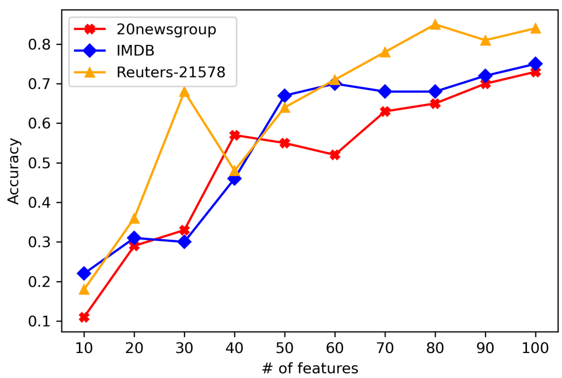

Fig. 5.4: The accuracy comparison of our proposed approach with respective to various number of features

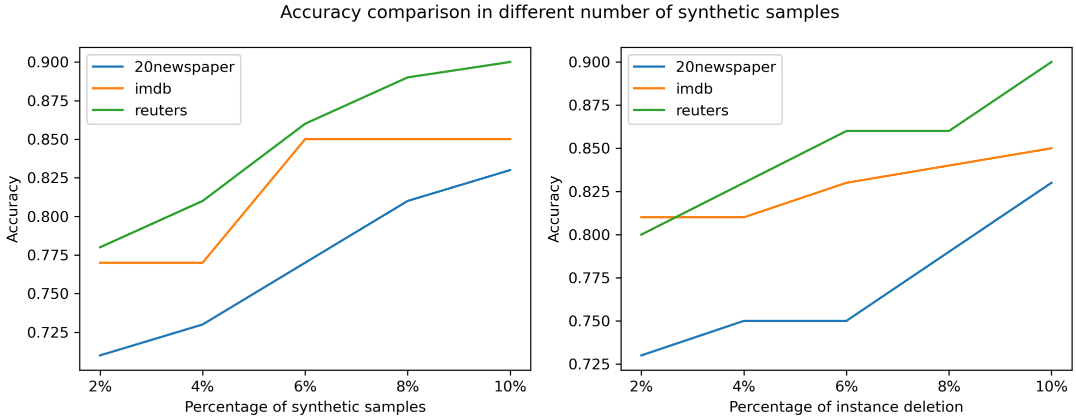

Fig. 5.5: The Accuracy Comparison in Imbalanced Dataset Tuning

The performance of the Reuters dataset is displayed in Table 5.5. The proposed classification model gives the best comparison result to the existing machine learning models. Naive Bayes produces the next highest accuracy and precision. The SVM produces the next highest recall.

From the Figures 5.1, 5.2 and 5.3, it can be observed that when the number of selected features increases, there is an increase in the performance of the machine learning model. When the number of selected features is fixed at 100, the relationship between the input features and the target class is not fully captured, and that's the reason for the low performance. Meanwhile, when the selected number of features increases, the machine learning model gets more information to learn about the relationship between the features and the target class and hence yields more accuracy. When the number of selected features is very low, the proposed approach is not producing satisfactory results. Figure 5.4 displays the accuracy when the number of selected features is less number of features. Moreover, we found that when the number of selected features exceeds 300, the classification performance for all three datasets becomes stable and no significant changes are found.

It is noted that the text classifiers perform much better in a balanced dataset than in an imbalanced dataset. There are two optimizations done to convert the imbalanced dataset into a balanced dataset. The first one is the creation of synthetic instances. Proper care is taken so that the created sample does not act as the original instance. The second one is the deletion of similar instances in the majority class without altering the original distribution of the dataset. Figure 5.5 displays the accuracy comparison in the two optimizations. The optimal value of both is 10%.

6. Conclusion and Future Work. Feature selection is one of the promising methods to increase the performance of the classification model. However, when the feature selection algorithm considers the entire class, then the selected features may not represent the unique characteristics of the classes. To solve this problem, class-specific features are selected so each class has its own set of feature subsets. This ensures that each subset fully represents the class and can hold the unique properties of the class. We propose a classification model that works on class-specific features and introduces ensembles on multiple levels. The experiment results prove that the proposed approach is effective and superior to the existing machine learning models. As the results are promising, the future scope of this research is planned to extend the work by incorporating deep learning approaches.

REFERENCES

[1] A. Palanivinayagam, C. Z. El-Bayeh, and R. Damaševičius, "Twenty years of machine-learning-based text classification: A systematic review," *Algorithms*, vol. 16, no. 5, 2023.
[2] J. Maldonado, M. C. Riff, and B. Neveu, "A review of recent approaches on wrapper feature selection for intrusion detection," *Expert Systems with Applications*, vol. 198, p. 116822, 2022.

[3] U. M. Khaire and R. Dhanalakshmi, "Stability of feature selection algorithm: A review," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 4, pp. 1060–1073, 2022.
[4] X. Liu, G. Zhou, M. Kong, Z. Yin, X. Li, L. Yin, and W. Zheng, "Developing multi-labeled corpus of Twitter short texts: A semi-automatic method," *Systems*, vol. 11, no. 8, 2023.
[5] S. Bazzaz Abkenar, M. Haghi Kashani, M. Akbari, and E. Mahdipour, "Learning textual features for Twitter spam detection: A systematic literature review," *Expert Systems with Applications*, vol. 228, p. 120366, 2023.
[6] L. Hu, L. Gao, Y. Li, P. Zhang, and W. Gao, "Feature-specific mutual information variation for multi-label feature selection," *Information Sciences*, vol. 593, pp. 449–471, 2022.
[7] G. S. Thejas, R. Garg, S. S. Iyengar, N. R. Sunitha, P. Badrinath, and S. Chennupati, "Metric and accuracy ranked feature inclusion: Hybrids of filter and wrapper feature selection approaches," *IEEE Access*, vol. 9, pp. 128687–128701, 2021.
[8] U. L. Altintakan and A. Yazici, "Towards effective image classification using class-specific codebooks and distinctive local features," *IEEE Transactions on Multimedia*, vol. 17, no. 3, pp. 323–332, 2015.
[9] C. P. Ezenkwu, U. I. Akpan, and B. U.-A. Stephen, "A class-specific metaheuristic technique for explainable relevant feature selection," *Machine Learning with Applications*, vol. 6, p. 100142, 2021.
[10] K. Liu, T. Li, X. Yang, X. Yang, and D. Liu, "Neighborhood rough set based ensemble feature selection with cross-class sample granulation," *Applied Soft Computing*, vol. 131, p. 109747, 2022.
[11] D. K. Rakesh and P. K. Jana, "A general framework for class label specific mutual information feature selection method," *IEEE Transactions on Information Theory*, vol. 68, no. 12, pp. 7996–8014, 2022.
[12] U. Upadhyay and B. Banerjee, "Compact representation learning using class specific convolution coders - application to medical image classification," in *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, pp. 1266–1270, 2020.
[13] S. Ruan, H. Li, C. Li, and K. Song, "Class-specific deep feature weighting for naïve bayes text classifiers," *IEEE Access*, vol. 8, pp. 20151–20159, 2020.
[14] F. Pan, Z.-X. Zhang, B.-D. Liu, and J.-J. Xie, "Class-specific sparse principal component analysis for visual classification," *IEEE Access*, vol. 8, pp. 110033–110047, 2020.
[15] M. Sewwandi, Y. Li, and J. Zhang, "A class-specific feature selection and classification approach using neighborhood rough set and k-nearest neighbor theories," *Applied Soft Computing*, vol. 143, p. 110366, 2023.
[16] R. Patil and V. M. Barkade, "Class-specific features using j48 classifier for text classification," in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, pp. 1–5, 2018.
[17] Z. Liang and H. Wang, "Efficient class-specific shapelets learning for interpretable time series classification," *Information Sciences*, vol. 570, pp. 428–450, 2021.
[18] A. Roy, P. D. Mackin, and S. Mukhopadhyay, "Methods for pattern selection, class-specific feature selection and classification for automated learning," *Neural Networks*, vol. 41, pp. 113–129, 2013. Special Issue on Autonomous Learning.
[19] X.-A. Ma, H. Xu, and C. Ju, "Class-specific feature selection via maximal dynamic correlation change and minimal redundancy," *Expert Systems with Applications*, vol. 229, p. 120455, 2023.
[20] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, p. 321–357, 2002.
[21] A. Zhang, H. Yu, S. Zhou, Z. Huan, and X. Yang, "Instance weighted smote by indirectly exploring the data distribution," *Knowledge-Based Systems*, vol. 249, p. 108919, 2022.
[22] A. Palanivinayagam and R. Damaševičius, "Effective handling of missing values in datasets for classification using machine learning methods," *Information*, vol. 14, no. 2, 2023.

Edited by: Polinpapilinho Katina
Special issue on: Scalable Dew Computing for Future Generation IoT systems
Received: Aug 10, 2023
Accepted: Sep 20, 2023